

**AMENDMENTS TO THE CLAIMS**

The following is a copy of Applicant's claims that identifies language being added with underlining ("\_\_") and language being deleted with strikethrough ("—"), as is applicable:

1. (Currently Amended) A computer readable storage medium having a program for automating the a life cycle of a grid node application, where the grid node application utilizes a plurality of computing resources distributed over a network, the program:

creating a file describing a plurality of stages of the life cycle, wherein the stages comprise development, packaging, distribution, installation, execution, collection, and uninstall task-list which describes how at least one stage in the life cycle of the grid node application is to be performed, wherein the task-list includes at least one of packaging all files needed to execute the grid node application into a single file, and the distribution stage includes at least distributing the grid node application to at least one remote computing resource on the network;

creating a task list in the file which describes how at least one of the stages in the life cycle of the grid node application is to be performed; and

processing the task list by a process engine to perform at least one stage in the life cycle,

wherein the process engine is integrated with a development environment, where the development environment is used to develop the grid node distributed computing software application.

2. (Original) The computer readable medium of claim 1, wherein the development environment is an integrated development environment.

3-4. (Cancelled)

5. (Currently Amended) The computer readable medium of claim 1, wherein the software grid node application utilizes computing resources through service providers connected to the network.

6. (Currently Amended) The computer readable medium of claim 1, wherein the file is task-list is stored in a text file.

7. (Previously Presented) The computer readable medium of claim 6, wherein the text file is an extensible markup language (XML) file.

8-9. (Cancelled)

10. (Currently Amended) The computer readable medium of claim 1, wherein the task-list includes a third task, wherein the third task execution stage includes executing executes the software grid node application on at least one remote computing resource.

11. (Currently Amended) The computer readable medium of claim 1, wherein the task-list includes a fourth task, wherein the fourth task collection stage includes collecting collects results from at least one remote computing resource.

12. (Currently Amended) The computer readable medium of claim 1, wherein the task-list includes a fifth task, wherein the fifth task removes uninstall stage includes removing the software grid node application from at least one remote computing resource.

13. (Currently Amended) A system embodied in a computer readable storage medium for automating the a life cycle of a software application, where the software application utilizes computing resources distributed over a network, the system comprising:

a task-list an editor configured to create a file task-list, where the file task-list describes how at least one step a plurality of stages in the life cycle of the software application, the stages

comprising development, packaging, distribution, installation, execution, collection, and uninstall, is to be executed, wherein the task list includes at least one of building the software application, packaging all files needed to run the software application into a single file, and wherein the distribution stage includes at least distributing the software application to at least one remote computing resource, wherein the software application utilizes computing resources distributed over the network when executing; and

wherein the file further includes a task list which describes how at least one of the stages in the life cycle of the software application is to be performed; and

a process engine configured to operate on the task list to perform the at least one stage step in the life cycle.

14. (Original) The system of claim 13, further comprising:

a development environment for developing the software application, where the process engine is integrated with the development environment.

15. (Original) The system of claim 14, wherein the development environment is an integrated development environment.

16. (Cancelled)

17. (Original) The system of claim 13, wherein the software application utilizes computing resources through service providers connected to the network.

18. (Currently Amended) The system of claim 13, wherein the file is task list is stored in a text file.

19. (Previously Presented ) The system of claim 18 wherein the text file is an extensible markup language (XML) file.

20-21. (Cancelled)

22. (Currently Amended) The system of claim 13, wherein the task-list includes a third task, wherein the third task executes execution stage includes at least executing the software application on at least one remote computing resource.

23. (Currently Amended) The system of claim 13, wherein the task-list includes a fourth task, wherein the fourth task collects collecting stage includes at least collecting results from at least one remote computing resource.

24. (Currently Amended) The system of claim 13, wherein the task-list includes a fifth task, wherein the fifth task removes uninstall stage includes removing the software application from at least one remote computing resource.

25. (Currently Amended) A system embodied in a computer readable storage medium for automating a the life cycle of a software application, where the software application utilizes computing resources distributed over a network, the system comprising:

creating logic operable to create a file describing a plurality of stages of the life cycle, wherein the stages comprise development, packaging, distribution, installation, execution, collection, and uninstall, task-list which describes how at least one stage in the application life cycle is to be performed, wherein the task-list includes at least one of building the software application, packaging all files needed to run the software application into a single file, and the distribution stage includes at least distributing the distributed computing software application to at least one remote computing resource; and

the file further comprising a task list which describes how at least one of the stages in the software application life cycle is to be performed; and

processing logic responsive to the creating logic, operable to process the task list to perform at least one stage in the software application life cycle,

wherein the processing logic is integrated with a development environment, wherein the development environment is used to develop the software application, ~~the software application utilizing computing resources distributed over the network when executing.~~

26. (Original) The system of claim 25, wherein the development environment is an integrated development environment.

27. (Cancelled)

28. (Original) The system of claim 25, wherein the software application utilizes computing resources through service providers connected to the network.

29. (Currently Amended) The system of claim 25, wherein the ~~file is task-list is stored in a text file.~~

30. (Previously Presented) The system of claim 29, wherein the text file is an extensible markup language (XML) file.

31. (Previously Presented) The computer readable medium of claim 1, wherein the processing the task list further comprises:

verifying that a precondition associated with a task in the task list is satisfied before performing the task.

32. (Currently Amended) The computer readable medium of claim 1, wherein the processing the task list further comprises:

verifying that a precondition is satisfied before performing the task, wherein the precondition is associated with a task in the task list and describes requirements of the system on which the ~~distributed-computing software grid node~~ application executes.

33. (Currently Amended) The computer readable medium of claim 1, wherein the processing the task list further comprises:

obtaining a description of available resources for at least a portion of the plurality of computing resources; and

verifying that a precondition is satisfied before performing a the task, wherein the precondition is associated with a task in the task list and describes the system requirements of the distributed computing software grid node application.

34. (Previously Presented) The system of claim 13, wherein the process engine is further configured to:

verify that a precondition associated with a task in the task list is satisfied before performing the task.

35. (Currently Amended) The system of claim 13, wherein the process engine is further configured to:

verify that a precondition is satisfied before performing a the task, wherein the precondition is associated with a task in the task list and describes requirements of the system on which the distributed computing software application executes.

36. (Currently Amended) The system of claim 13, wherein the process engine is further configured to:

obtain a description of available resources for at least a portion of the plurality of computing resources; and

verify that a precondition is satisfied before performing a the task, wherein the precondition is associated with a task in the task list and describes the system requirements of the distributed computing software application.

37. (Previously Presented) The system of claim 25, wherein the processing logic is further configured to:

verify that a precondition associated with a task in the task list is satisfied before performing the task.

38. (Currently Amended) The system of claim 25, wherein the processing logic is further configured to:

verify that a precondition is satisfied before performing a the task, wherein the precondition is associated with a task in the task list and describes requirements of the system on which the distributed computing software application executes.

39. (Currently Amended) The system of claim 25, wherein the processing logic is further configured to:

obtain a description of available resources for at least a portion of the plurality of computing resources; and

verify that a precondition is satisfied before performing a the task, wherein the precondition is associated with a task in the task list and describes the system requirements of the distributed computing software application.